UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/757,764	01/09/2001	Daniel J. Scales	9772-0268-999	8427

24341 7590 06/17/2004

MORGAN, LEWIS & BOCKIUS, LLP.
3300 HILLVIEW AVENUE
PALO ALTO, CA 94304

EXAMINER	
YIGDALL, MICHAEL J	

ART UNIT	PAPER NUMBER
2122	5

DATE MAILED: 06/17/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/757,764	SCALES, DANIEL J.	
	Examiner	Art Unit	
	Michael J. Yigdall	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 April 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is in reply to Applicant's response and amendment dated April 1, 2004. Claims 1-39 are now pending.

Response to Arguments

2. Applicant's arguments with respect to claims 1-36 have been considered but are moot in view of the new ground(s) of rejection.

Applicant contends that neither Bharadwaj nor Moreno suggest or disclose generating a summary for each memory operation in the graph representation that indicates for each location type used in the procedure the closest preceding memory operation to affect that location type, as recited in amended independent claims 1, 13 and 25 (see page 16).

However, Ruf teaches a system for type-partitioned data flow analysis (see the title and abstract). As set forth below, Ruf discloses building a dependence graph that indicates the data or location type of each node and serves as a summary for determining the ordering of the nodes, i.e. the ordering of memory operations, based on type (see column 9, lines 40-49, and column 9, line 66 to column 10, line 21). The dependence graphs, which take the form of directed acyclic graphs, would indicate for each node or memory operation the closest preceding node or memory operation associated with that type.

Claim Rejections - 35 USC § 103

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 1, 2, 5, 7-14, 17, 19-26, 29 and 31-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,758,051 to Moreno et al. (hereinafter "Moreno") in view of U.S. Pat. No. 5,787,287 to Bharadwaj in view of U.S. Pat. No. 6,077,313 to Ruf.

With respect to claim 1 (currently amended), Moreno discloses a method for modifying serial dependencies in a procedure (see column 9, lines 38-45).

Although Moreno discloses reordering memory operations and scheduling the associated instructions (see FIG. 2A), Moreno does not expressly disclose the steps of:

(a) building a graph representation of said procedure, said graph representation having an origin and including a unique position, relative to said origin, for each memory operation in said procedure;

(b) designating a location type for each memory operation in said graph representation; each said location type based on a characteristic of said corresponding memory operation;

However, Bharadwaj discloses step (a) above in terms of building vectors to represent data dependency and control flow paths for instructions in a procedure (see FIG. 3 and column 3, lines 32-35; see also FIG. 2 and column 3, lines 50-56, which shows a control flow diagram or graph having an origin and a unique position for each memory operation).

Bharadwaj further discloses step (b) above in terms of designating each memory operation according to its associated variable, i.e. its data type, and classifying each instruction as a reader or a writer based on its characteristics (see column 4, lines 4-25).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Bharadwaj, for the

purpose of determining whether reordering memory operations affects the meaning of the source code (see Bharadwaj, column 6, lines 40-43).

Moreno does not expressly disclose the step of:

(c) generating a summary for each memory operation in the graph representation that indicates for each location type used in the procedure the closest preceding memory operation to affect the location type;

However, Ruf discloses building a dependence graph to represent each location type used in a procedure (see column 9, lines 40-49) and using the graph, a summary indicating the location type of each node, to determine the ordering of types and dependencies with data flow analysis (see column 9, line 66 to column 10, line 21). Note that the ordering would indicate for each node or memory operation the closest preceding node or memory operation for that type.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Ruf, for the purpose of partitioning the data flow analysis by location type to minimize the execution time and memory space requirements (see Ruf, column10, lines 31-58).

Bharadwaj further discloses the step of:

(d) identifying a first memory operation having the same location type as a second memory operation; wherein said first memory operation is positioned closer to said origin than said second memory operation, and said graph representation does not include any additional memory operations of the same location type between the first and second memory operations (see column 5, lines 3-5, which shows computing the dependency path vector between two instructions that operate on the same variable, i.e. that have the same location type; see also

column 4, lines 52-58, which shows that one memory operation comes before the other, i.e. that one memory operation is closer to the origin, and column 5, lines 5-10, which shows that no related instructions are included between the two memory operations).

Moreno further discloses the step of:

(e) moving said second memory operation to a new position in said graph representation that is closer to said first memory operation (see column 9, lines 55-67, which shows moving a second memory operation to an earlier position, i.e. closer to a first memory operation).

With respect to claim 2 (original), although Moreno discloses moving memory operations in the schedule of instructions, thereby modifying serial dependencies (see column 9, lines 38-45), Moreno does not expressly disclose the limitations wherein:

(a) the building step includes the step of assigning an initial set of serial dependencies between program operations represented in said graph representation;

(b) the moving step includes (i) removing one or more of the serial dependencies in said initial set of serial dependencies that is associated with said second memory operation and (ii) creating a new serial dependency between said first memory operation and said second memory operation.

However, Bharadwaj further discloses limitation (a) above in terms of computing an initial set of dependency path vectors between instructions (see column 6, lines 51-54).

Bharadwaj further discloses limitation (b) above in terms of moving code, which removes serial dependencies, and computing a new set of dependency path vectors (see column 6, lines 55-66).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Bharadwaj, for the purpose of determining whether reordering memory operations affects the meaning of the source code (see Bharadwaj, column 6, lines 40-43).

With respect to claim 5 (original), Moreno further discloses the limitation wherein said moving step results in said second memory operation being advanced in a schedule of machine instructions (see column 9, lines 55-67, which shows moving a second memory operation to an earlier position, i.e. advancing the operation in the schedule of instructions).

With respect to claim 7 (original), Moreno further discloses the limitation wherein said first memory operation is a store or an array store and said second memory operation is a load or an array load (see column 9, lines 63-67, which shows the case wherein the second memory operation is a load and the first memory operation is a store).

With respect to claim 8 (original), Moreno further discloses the limitation wherein said procedure includes a calling procedure and said first memory operation is in said calling procedure and said second memory operation is in a called procedure that is called by an operation in said calling procedure (see column 9, lines 33-48, which shows scheduling all the instructions in a program, including those associated with a procedure called by another procedure; note that a calling procedure may include a first memory operation and a called procedure may include a second memory operation).

With respect to claim 9 (original), Moreno further discloses the limitation wherein said moving step results in a placement of said second memory operation in said calling procedure (see column 9, lines 33-48, which shows moving a second memory operation to an earlier position in the schedule of instructions; note that this applies to all the instructions in a program, and could result in moving a second memory operation from a called procedure to a calling procedure).

With respect to claim 10 (original), Moreno does not expressly disclose the limitation wherein said location type of each memory operation is selected from the group consisting of a predefined set of base types, a predefined set of array types, object types, and object field types.

However, Bharadwaj further discloses the limitation above in terms of identifying memory operations that reference a particular variable, i.e. memory operations associated with the same data type (see column 4, lines 4-15; note that the data type or location type of a variable is necessarily selected from a predefined set of base types and array types, or from a set of object types and object field types defined in the code).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Bharadwaj, for the purpose of determining whether reordering memory operations affects the meaning of the source code (see Bharadwaj, column 6, lines 40-43).

With respect to claim 11 (original), Moreno does not expressly disclose the limitation wherein the building step further comprises adding a global store dependency to each operation in said procedure that reads a variable from or stores a variable to memory.

However, Bharadwaj discloses computing dependency path vectors for each instruction that reads or writes a variable (see column 4, lines 4-15; note that a store dependency would be added to each operation following a store instruction that references the same variable).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Bharadwaj, for the purpose of determining whether reordering memory operations affects the meaning of the source code (see Bharadwaj, column 6, lines 40-43).

Moreno further discloses the step of generating a schedule of machine instructions in accordance with said graph representation, wherein each said machine instruction in said schedule of machine instructions, which corresponds to an operation that reads a variable from or stores a variable to memory, is ordered in accordance with said global store dependency associated with said operation (see FIG. 1, which shows generating a schedule of instructions, and column 9, lines 59-67, which shows verifying that memory operations are ordered in accordance with a store dependency).

With respect to claim 12 (original), Moreno does not expressly disclose the limitation wherein a first operation affects a value of a variable stored in memory and a second operation serially follows said first operation, said building step further comprising adding a global store dependency from said second operation to said first operation.

However, Bharadwaj discloses the limitation above in terms of computing dependency path vectors for each instruction that reads or writes a variable (see column 4, lines 4-15; note that a store dependency would be added from a second operation to a first operation when both instructions reference the same variable).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Bharadwaj, for the purpose of determining whether reordering memory operations affects the meaning of the source code (see Bharadwaj, column 6, lines 40-43).

Moreno further discloses the step of generating a schedule of machine instructions in accordance with said graph representation, wherein said machine instructions in said schedule of machine instructions corresponding to said second operation are scheduled after said machine instructions corresponding to said first operation (see FIG. 1, which shows generating a schedule of instructions, and column 10, lines 50-60, which shows ensuring that instructions are executed in the order prescribed by the dependencies).

With respect to claim 13 (currently amended), see the explanation for analogous claim 1 above. Note that Moreno further discloses a computer program product for use in a computer system (see column 1, lines 18-22).

With respect to claim 14 (original), see the explanation for analogous claim 2 above.

With respect to claim 17 (original), see the explanation for analogous claim 5 above.

With respect to claim 19 (original), see the explanation for analogous claim 7 above.

With respect to claim 20 (original), see the explanation for analogous claim 8 above.

With respect to claim 21 (original), see the explanation for analogous claim 9 above.

With respect to claim 22 (original), see the explanation for analogous claim 10 above.

With respect to claim 23 (original), see the explanation for analogous claim 11 above.

With respect to claim 24 (original), see the explanation for analogous claim 12 above.

With respect to claim 25 (currently amended), see the explanation for analogous claim 1 above. Note that Moreno further discloses a computer system comprising a memory and a processor (see FIGS. 3 and 5).

With respect to claim 26 (original), see the explanation for analogous claim 2 above.

With respect to claim 29 (original), see the explanation for analogous claim 5 above.

With respect to claim 31 (original), see the explanation for analogous claim 7 above.

With respect to claim 32 (original), see the explanation for analogous claim 8 above.

With respect to claim 33 (original), see the explanation for analogous claim 9 above.

With respect to claim 34 (original), see the explanation for analogous claim 10 above.

With respect to claim 35 (original), see the explanation for analogous claim 11 above.

With respect to claim 36 (original), see the explanation for analogous claim 12 above.

With respect to claims 37-39 (new), Moreno does not expressly disclose the limitation wherein no known preceding memory operation in the procedure affects a location type, said summary indicates the origin for that location type.

However, Ruf discloses building a dependence graph to represent each location type used in a procedure (see column 9, lines 40-49) and using the graph, a summary indicating the location type of each node, to represent the ordering of types and dependencies in a directed acyclic graph (see column 9, line 66 to column 10, line 21). Note that a node for which no known preceding memory operation affects its location type will be indicated as an origin for that location type in the ordered graph.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Ruf, for the purpose of partitioning the data flow analysis by location type to minimize the execution time and memory space requirements (see Ruf, column10, lines 31-58).

5. Claims 3, 4, 15, 16, 27 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moreno in view of Bharadwaj in view of Ruf as applied to claims 1, 13 and 25 above, respectively, and further in view of U.S. Pat. No. 6,016,398 to Radigan (hereinafter "Radigan 398").

With respect to claim 3 (original), Moreno does not expressly disclose the limitation wherein said graph representation is an intermediate representation.

However, Radigan 398 discloses the limitation above in a system for removing dependencies (see FIG. 2A, which shows creating an intermediate representation of a program comprising nodes for a graph), in order to ensure that each variable expression will be reached by at most one definition (see column 6, lines 56-61).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Radigan 398, for the purpose of ensuring the consistency of variable definitions.

With respect to claim 4 (original), Moreno does not expressly disclose the limitation wherein said intermediate representation is a single static assignment graph embedded in a control flow graph.

However, Radigan 398 further discloses the limitation above (see FIG. 4A, which shows determining the flow of control along each execution path, and FIG. 3A and column 14, lines 14-16, which shows a corresponding graph; see also column 6, lines 56-61, which shows using static single assignment or SSA in order to ensure that each variable expression will be reached by at most one definition).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno and Bharadwaj with the features taught by Radigan 398, for the purpose of ensuring the consistency of variable definitions.

With respect to claim 15 (original), see the explanation for analogous claim 3 above.

With respect to claim 16 (original), see the explanation for analogous claim 4 above.

With respect to claim 27 (original), see the explanation for analogous claim 3 above.

With respect to claim 28 (original), see the explanation for analogous claim 4 above.

6. Claims 6, 18 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moreno in view of Bharadwaj in view of Ruf as applied to claims 1, 13 and 25 above, respectively, and further in view of U.S. Pat. No. 6,151,704 to Radigan (hereinafter "Radigan 704").

With respect to claim 6 (original), Moreno does not expressly disclose the limitation wherein said first memory operation is positioned before a repetitive loop in said procedure and said second memory operation is within said repetitive loop; said graphic representation including a phi node that corresponds to a loop back position in said repetitive loop.

However, Radigan 704 discloses the limitation above for the purpose of optimizing a loop in a computer program (see column 6, lines 3-11, which shows a memory operation within a loop and a memory operation prior to the loop; see also column 4, lines 32-36, which shows the use of phi functions or phi nodes to denote a merge point, such as a loop back position).

Radigan 704 further discloses the limitations wherein:

(a) when a memory operation having the same location type as said first and second memory operation exists in said loop, said new position in said graph representation is a position that is serially dependent upon said phi node (see column 6, lines 3-11, which shows determining whether a memory operation writes to a variable referenced by the first and second memory operations, and thus has the same location type and is dependent upon the phi node); and

(b) when a memory operation having the same location type as said first and second memory operation does not exist in said loop, said new position in said graph representation is a position that is not serially dependent on any operation in the loop (see column 6, lines 3-11,

which shows determining whether a memory operation is not within the loop and thus is not dependent upon instructions in the loop).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Moreno with the features taught by Radigan 704, for the purpose of optimizing loops in a computer program.

With respect to claim 18 (original), see the explanation for analogous claim 6 above.

With respect to claim 30 (original), see the explanation for analogous claim 6 above.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (703) 305-0352. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall
Examiner
Art Unit 2122

mjy
June 1, 2004

TUAN DAM
SUPERVISORY PATENT EXAMINER